

Galaxy Game – Space Cowboys

TIE-02401 Ohjelmoinnin tekniikat Syksy 2017

Petri Leppänen, petri.leppanen@student.tut.fi, 160280

Janne Uitto, janne.uitto@student.tut.fi, 276150

1. Luokkien vastuujako

class CustomScene: public QGraphicsScene

CustomScene hallitsee piirrettäviä asioita starsystemeitä, laivoja sekä pelaajaa. Tämän lisäksi scene kuuntelee hiiri ja näppäimistö klikkauksia pelissä, joiden perusteella pelaaja pelaa peliä.

class StarSystemDraw: public QObject, public QGraphicsPixmapItem

StarSystemDraw kuvaa yhtä starsystem ikonia scenessä, tämä StarSystemDraw myös sisältää starsystemissä olevat laivat, jotka on piirretty pienillä neliöillä starsystemin päälle. Tässä luokassa on esitelty jälkiehtoja.

class EventHandler: public Common::IEventHandler

EventHandler –luokka joka on periytynyt kurssin IEventHandlerista. Handlerin saamien eventtien perusteella piirretään sceneen starsystemeitä uudelleen, riippuen mistä liikkuva laiva lähtee ja tulee. Handlerissa myös seurataan pelaajan pelastamia laivoja sekä kadotettuja laivoja.

class Galaxy : public Common::IGalaxy, public std::enable_shared_from_this<Galaxy>

Galaxy –luokka joka on periytynyt kurssin IGalaxysta. Luokan toiminta toteutettu lähinnä kurssin yksikkötestien perusteella.

class startingdialog : public QDialog

Startingdialogia avautuu ohjelman käynnistyessä ja välittää signaalin pelaaja-alusten määrästä pelin MainDialogiin, jos aloitetaan uusi peli. Startingdialog avautuu myös pelin loputtua, jossa nyt näkyy signal-slotin avulla maindialogista edellisen pelin pistemäärä. Pelin jälkeen startingdialogista voi taas aloittaa uuden pelin.

class MainDialog: public QDialog

MainDialog –ikkunassa varsinainen peli lähtee käyntiin Startingdialogin jälkeen. MainDialogissa on scene ja pelin etenemistä kertovat tilastot pelastetuista ja kadotetuista laivoista. Dialogi hallitsee pelin kelloa, sekä tarkistaa kun peli loppuu liian moneen kadotettuun laivaan. Kellon avulla tehdään laivojen actioneita.

class Player: public QObject, public QGraphicsPixmapItem, public Common::Ship

Pelaaja –luokka. Lähinnä vain ui elementti, toteutuksessa pelaajalla omia actioneita. Hätähuudon pyytäneitä laivoja pelastetaan menemällä starsystemin lähelle, jossa on tällainen laiva. Pelaajaa siirretään hiiren klikkauksella.

class Statistics : public Common::IStatistics

Statistics –luokka joka on periytynyt kurssin IStatisticsta. Statistiikkaa peliin, jossa tallessa pelin eri saavutuksia kuten pelastettujen ja kadonneiden alusten määrä.

2. Ohjelman toiminta

Pelin ideana on pelastaa hätähuudon antaneita rahtialuksia. Rahtialuksia pelastetaan klikkaamalla pelaaja-alus tarpeeksi lähelle tähtijärjestelmää, jossa hätähuuto-alus on, tällöin aluksen korjaaminen alkaa. Korjaamiseen kuluva aika riippuu rahtialuksen viasta. Peli pyörii ns. reaaliaikaisesti kellon avulla, jossa ns. yksi vuoro päivittyy 2s välein, jolloin lisätään yksi alus peliin ja kaikille rahtialuksille pyydetään ja suoritetaan toiminto.

Pelissä saa aina yhden pisteen, kun onnistuu rahtialuksen pelastamisessa. Jos hätähuudon tehnyt alus ei saa apua pitkään aikaan se hylätään. Peli loppuu, kun rahtialuksia on hylätty yhteensä 10 kappaletta.

Ohjelman avautuessa, pelaaja voi määritellä ensimmäisessä ikkunassa, montako rahtialusta tähtijärjestelmään luodaan varsinaisen pelin aloituksessa. Tällä asetuksella voi hakea vähän ns. vaikeusastetta kuinka haastavasta tilanteesta peli lähtee alkaa.

Alukset piirtyvät turkoosin värisillä neliöillä, jos ne ovat Galaxyssa. Hätähuudon antaneet aluksien väri muuttuu punaiseksi.

Pelaaja-alusta voi siirtää hiirellä paikasta toiseen välittömästi. Pelaaja-alusta voi siirtää myös WASD-näppäimillä sekä karttaa voi zoomata E- ja Q- näppäimillä.

3. Lisäominaisuudet

Osa listatuista asioista (kuin myös osa vaadituista asioista) oli aika paljon tulkinnan varaisia. Ohessa lyhyet selvitykset mitkä toiminnot ovat meidän harjoitustyössä mukana.

Peli toimii reaaliaikaisesti.

Pelissä on kello, joka juoksee taustalla kahden sekunnin välein. Kellon avulla npc-alukset saavat toimintoja ja suorittavat niitä.

Minimaalinen ruudunpäivitys.

(Toimijan tilan muutoksen seurauksena vain sen välitön ympäristö päivitetään pelialueelle, sen sijaan että koko kartta piirrettäisiin uudelleen.)

EventHandlerin perusteella piirretään uudelleen vain tarvittava starsystem, ja sen laivat.

Vieritettävä kartta.

(Pelialue ei ole sidottu rajattuun näkymään, vaan esitettävä alue vaihtuu pelaajan aluksen liikkeiden perusteella.)

WASD-näppäimillä voi siirtää karttaa, sekä Q- ja E-näppäimillä voi zoomata karttaa.

Itse toteutetut yksikkötestit CI-ympäristössä

CI:ssä on jobi, joka ajetaan joka pushin jälkeen. CI runnerista puuttuu qmake-qt5 (/qt4) työkalu, jotta pystyisi qt-sovelluksen make tiedoston ja käännöksen tekemään.

4. Työnjako

Projektin alussa vedettiin yhteen mitä aletaan tehdä ja millaisin tavoittein. Tavoitteena meillä oli perustoteutus, mutta tästä jäätin hieman vajaaksi (tähtien tietojen saaminen näkyviin käyttöliittymään). Muutamia lisäominaisuuksia tuli lisäksi vähän kompensoimaan tolppaa.

Pelin logiikasta suurin osa on Jannen tekemää mm. Galaxy, Scene sisältäen klikkaukset sekä piirrettävät asiat, MainDialog, EventHandlerin kytkeminen muuhun pelin tarvittaviin osiin. Petri on toteuttanut Statistics luokan ja sen testit sekä StartingDialogista suuremman osuuden.